#100 DAYS OF RTL

https://github.com/Bhuvan-2602 www.LinkedIn.com/bhuvan-p-4825a9358

Description:

An **8-bit Multiplexer (8:1 MUX)** selects one of eight input lines and forwards it to a single output based on the values of three select lines s1, s2, and s3.

Objective:

Understand how to implement an 8-to-1 multiplexer using basic logic gates, and how selection is made using binary-coded select lines.

Inputs:

- d: 8-bit data input (d[0] to d[7])
- s1, s2, s3: 1-bit select lines (3 bits total → 8 combinations)

Outputs:

- out: 1-bit selected output

Design Approach:

- Use 3 select lines to choose one of 8 input bits.
- Implement the logic using AND/OR expressions corresponding to binary decoding.
- Only the selected data line (based on s1s2s3) will pass through to the output.

Verilog Code:

8-bit Multiplexer

```
module eb_mux(
   input [7:0] d,
   input s1, s2, s3,
   output out
);
   assign out = (d[0] & ~s1 & ~s2 & ~s3) |
```

```
            (d[1] & ~s1 & ~s2 &  s3) |
            (d[2] & ~s1 &  s2 & ~s3) |
            (d[3] & ~s1 &  s2 &  s3) |
            (d[4] &  s1 & ~s2 & ~s3) |
            (d[5] &  s1 & ~s2 &  s3) |
            (d[6] &  s1 &  s2 & ~s3) |
            (d[7] &  s1 &  s2 &  s3);
endmodule
```

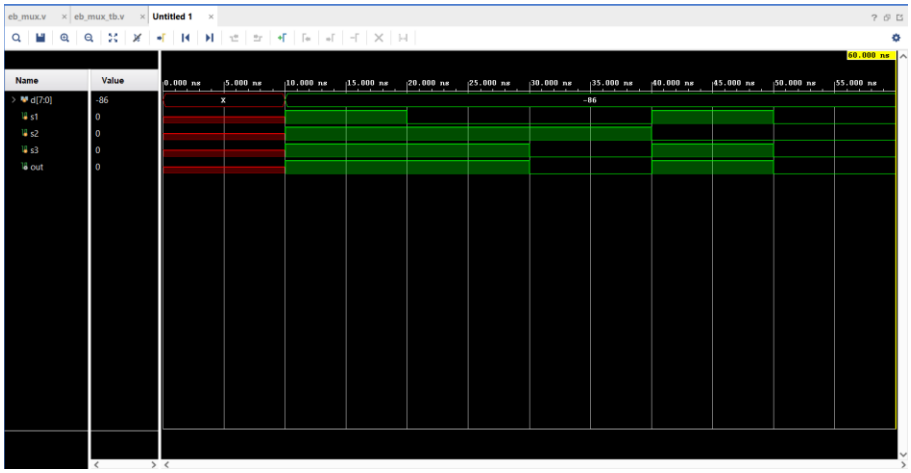
Testbench:

```
module eb_mux_tb;
   reg [7:0] d;
   reg s1, s2, s3;
   wire out;

   eb_mux dut (d, s1, s2, s3, out);

   initial begin
      #10;
      s1 = 1'b1; s2 = 1'b1; s3 = 1'b1; d = 8'b10101010;  // Select d[7]
      #10;
      s1 = 1'b0; s2 = 1'b1; s3 = 1'b1; d = 8'b10101010;  // Select d[3]
      #10;
      s1 = 1'b0; s2 = 1'b1; s3 = 1'b0; d = 8'b10101010;  // Select d[2]
      #10;
      s1 = 1'b1; s2 = 1'b0; s3 = 1'b1; d = 8'b10101010;  // Select d[5]
      #10;
      s1 = 1'b0; s2 = 1'b0; s3 = 1'b0; d = 8'b10101010;  // Select d[0]
      #10;
      $finish();
   end
endmodule
```

Waveform:



Schematic: